REMARKS

In the Office Action, the Examiner rejected claims 1-19, 21, 22, and 26-29 under 35 USC §102. In addition, the Examiner indicated that claims 20 and 23-25 are objected to. A copy of the drawings is submitted herewith. The claims have been amended to correct typographical errors and further clarify the subject matter regarded as the invention. The claim rejections are fully traversed below.

Reconsideration of the application is respectfully requested based on the following remarks.

REJECTION OF CLAIMS UNDER 35 USC §102

In the Office Action, the Examiner rejected the claims under 35 USC §102 as being anticipated by Judge et al, U.S. Patent No. 6,430,564, ('Judge' hereinafter). This rejection is fully traversed below.

Judge discloses a data manager for managing global data within a Java Virtual Machine (JVM). See Abstract. The data manager includes a data class loader method, a data object creation method, get and put data methods which allow manipulation of existing data objects, and an unload method which unloads cached data objects from the embedded device. The data manager comprises or is responsive to a memory management handler which detects low- or out-of-memory conditions and which selects one or more data class objects to be unloaded from the data cache. See Abstract.

In contrast, the present invention provides an application manager for managing execution of an application. The Examiner cites Judge, indicating that the pending claims are anticipated by Judge. However, Applicant respectfully submits that Judge fails to anticipate the pending claims, as will be discussed in further detail below.

With respect to claim 1, Judge fails to disclose "A state machine for an application manager that manages execution of an application in a digital television receiver environment, said state machine comprising:

a loaded state in which the application has been loaded;". The Examiner cites col. 3, lines 1-7. However, col. 3, lines 1-7 indicate that a client application sends requests (e.g., unloadData Class Data1...)...to embedded device 20, which operates as a server to service the requests. In other words, an application loads a data class. Thus, Judge fails to disclose a state machine including a loaded state in which the application has been loaded. While col. 3, line 57 indicates that the application manager provides downloading, starting, stopping, querying, and memory management capabilities, Judge fails to disclose or suggest a paused state in which the application is paused, the application being initialized to transition from said loaded state to said paused state. In addition, Judge fails to disclose a state machine including an active state in which the application is executing, the application being started to transition from said paused state to said active state. Moreover, Judge fails to disclose or suggest a destroyed state in which the application is destroyed, the application being terminated to transition from either said active state or said paused state to said destroyed state. While the Examiner cites col. 5, lines 34-35, Judge merely discloses that a data object is destroyed, not the application. Accordingly, Applicant respectfully requests that the Examiner withdraw the rejection of claim 1.

Claim 2 recites "A state machine as recited in claim 1, wherein the application can transition from said loaded state to said destroyed state when the application is to be terminated while in said loaded state." Col. 5, lines 40-45 indicate that data classes are unloaded by the data manager, not that the application is unloaded. Thus, Applicant respectfully submits that claim 2 is patentable over the cited art.

Claim 3 recites "A state machine as recited in claim 2, wherein either of the application manager or the application can initiate the transition to said destroyed state." The Examiner cites col. 6, lines 48-53 of Judge. However, Judge merely indicates that a memory management handler enables unloaded classes to be reclaimed. Judge fails to disclose or suggest that either the application manager or the application can initiate the transition to the destroyed state. Accordingly, Applicant respectfully submits that claim 3 is allowable over the cited art.

With respect to claims 4-6, col. 3, lines 56-57 indicate that the application manager provides downloading, starting, stopping, querying, and memory management capabilities. However, as recited in claim 5, Judge is silent as to whether the application can initiate the transition from the active to the paused state. Accordingly, Applicant respectfully submits that claim 5 is allowable over the cited art.

With respect to claim 7, Judge fails to disclose the states of a state machine as claimed and the specified transitions which together form an application lifecycle. Accordingly, Applicant respectfully submits that claim 7 is allowable over the cited art.

Claim 8 recites: "A computer program product for managing execution of an application according to an application lifecycle, the computer program product comprising:

a computer-readable medium storing computer-readable instructions thereon, the computer-readable instructions including:

instructions for loading the application such that the application enters a loaded state;

instructions for initializing the application when the application is in the loaded state such that the application enters a paused state;

instructions for starting execution of the application when the application is in the paused state such that the application enters an active state; and

instructions for terminating the execution of the application when the application is in the loaded state, the paused state, or the active state such that the application enters a destroyed state."

Judge fails to disclose "loading the application such that the application enters a loaded state." The Examiner cites col. 3, lines 1-7 of Judge. However, Judge merely discloses the loading of a data class, <u>not the application</u>. Moreover, Judge fails to disclose or suggest the state transitions as claimed. For instance, Judge fails to disclose or suggest "terminating the execution of the application when the application is in the loaded state, the paused state, or the active state such that the application enters a destroyed state." The Examiner cites col. 5, lines 34-35 of Judge. However, Judge merely indicates that the "data object" is destroyed, <u>not the application</u>. Accordingly, Applicant respectfully submits that claim 8 is allowable over the cited art.

Claim 9 recites "pausing the application when the application is in the active state such that the application enters the paused state." Col. 5, lines 40-45 indicate that data classes are unloaded by the data manager, <u>not that the application is unloaded or paused</u>. Accordingly, Applicant respectfully submits that claim 9 is allowable over the cited art.

Claim 12 recites: "The computer program product as recited in claim 9, wherein the instructions for pausing the execution of the application when the application is in the active state can be called by the application or a process external to the application." The Examiner cites col. 6, lines 43-46 of Judge, which indicates that application or data objects objects may be dumped from memory. However, Judge fails to disclose or suggest that <u>the application itself may choose to pause its execution</u>. Accordingly, Applicant respectfully submits that Judge fails to anticipate claim 12.

Claim 13 recites "The computer program product as recited in claim 8, wherein the instructions for terminating the application can be executed by the application or a process external to the application." The Examiner cites col. 6, lines 49-55 of Judge, which indicates that the memory management capabilities includes informing the JVM that unloaded class objects and instances of the class objects are available to be reclaimed. Judge fails to disclose or suggest that the application can terminate itself by executing the instructions for terminating the application. Accordingly, Applicant respectfully submits that Judge fails to anticipate claim 13.

Claim 14 recites: "A computer program product for managing execution of an application according to an application lifecycle, the computer program product comprising:

a computer-readable medium storing computer-readable instructions thereon, the computer-readable instructions including:

instructions for initializing an application such that the application enters a paused state;

instructions for starting execution of the application such that the application enters an active state;

instructions for pausing the execution of the application such that the application enters the paused state; and

instructions for terminating the application such that the application enters a destroyed state."

Judge fails to disclose "initializing an application such that the application enters a paused state." The Examiner cites col. 5 lines 12-15 and Appendix A/B of Judge. However, col. 5, lines 12-15 and Appendix A/B of Judge relates to the initialization of a data object, not an application. Moreover, Judge fails to disclose that the application enters a paused state. While the Examiner cites col. 5, lines 34-35, Judge merely discloses that a data object is destroyed, not the application. Thus, Judge fails to disclose "terminating the application such that the application enters a destroyed state." Accordingly, Applicant respectfully submits that claim 14 is allowable over the cited art.

Claim 15 recites "A computer program product for managing execution of an application according to an application lifecycle, the computer program product comprising:

a computer-readable medium storing computer-readable instructions thereon, the computer-readable instructions including:

instructions for starting execution of the application such that the application enters an active state;

instructions for pausing the execution of the application such that the application enters the paused state;

instructions for conditionally terminating the execution of the application such that the application enters a destroyed state when a predetermined condition is satisfied; and

instructions for unconditionally terminating the execution of the application such that the application enters the destroyed state when the predetermined condition is not satisfied."

While the Examiner cites col. 5, lines 34-35, Judge merely discloses that a data object is destroyed, not the application. As such, Judge fails to disclose conditional termination of the application as the Examiner suggests. Since, Judge fails to disclose or suggest conditional and unconditional termination of the application, Applicant respectfully submits that claim 15 is allowable over the cited art.

Claim 16 recites: "The computer program product as recited in claim 15, wherein the predetermined condition is a signal received from the application." The Examiner cites col. 5, lines 37-39 of Judge, which indicates that the memory management handler selects the data object to be unloaded as a result of a low- or no-memory condition. However, Judge fails to disclose that conditional and unconditional termination is triggered by a signal received from the application being terminated. Accordingly, Applicant respectfully submits that claim 16 is allowable over the cited art.

Claim 17 recites: "The computer program product as recited in claim 15, wherein the predetermined condition is an absence of a signal received from the application within a specified period of time." The Examiner cites col. 8, lines 9-15 of Judge, which indicates that when it is determined that data class objects or instances of data objects are to be unloaded due to a low- or no-memory situation, the Data Manager removes references to the objects that are chosen to be unloaded. Thus, Judge fails to disclose or suggest that conditional and unconditional termination is triggered by the absence of a signal received from the application being terminated within a specified period of time. Accordingly, Applicant respectfully submits that claim 17 is allowable over the cited art.

Claim 18 recites: "The computer program product as recited in claim 15, further comprising: instructions for ignoring a state change exception raised by the application when the predetermined condition is not satisfied, the state change exception indicating that the application does not want to terminate." The Examiner cites col. 7, lines 33-44 of Judge, which indicates that the data manager receives a request from an application, which may result in the use of free memory. A determination is made as to whether the execution of the received request would result in a low- or no-memory condition, which may result in the unloading of application or data objects. Judge neither discloses nor suggests "ignoring a state change exception raised by the application when the predetermined condition is not satisfied, the state change exception indicating that the application does not want to terminate." Accordingly, Applicant respectfully submits that claim 18 is allowable over the cited art.

Claim 19 recites: "A computer program product for managing execution of an application according to an application lifecycle, the computer program product comprising:

a computer-readable medium storing computer-readable instructions thereon, the computer-readable instructions including:

instructions for starting execution of the application such that the application enters an active state;

instructions for pausing the execution of the application such that the application enters the paused state;

instructions for terminating the application such that the application enters a destroyed state; and

an interface including a set of instructions that enable a process other than the application to initiate execution of the instructions for starting execution of the application, <u>that enable a process other than the application or the application to initiate execution of</u> the instructions for pausing the execution of the application, and <u>that enable a process other than the application or the application to initiate execution of</u> the instructions for terminating the application.

Judge fails to disclose or suggest "an interface including a set of instructions that enable a process other than the application to initiate execution of the instructions for starting execution of the application, <u>that enable a process other than the application or the application to initiate execution of</u> the instructions for pausing the execution of the application, and <u>that enable a process other than the application or the application to initiate execution of</u> the instructions for terminating the application." In other words, the application as claimed may initiate its termination or pause its execution. Accordingly, Applicant respectfully submits that claim 19 is patentable over the cited art.

Claim 21 recites: "A computer program product for managing execution of an application according to an application lifecycle, the computer program product comprising:

a computer-readable medium storing computer-readable instructions thereon, the computer-readable instructions including:

instructions for communicating that the application has decided to terminate and has entered a destroyed state from a loaded state, a paused state, or an active state; and

instructions for communicating that the application has decided to pause its execution and has entered the paused state from the active state."

Judge neither discloses nor suggests the claimed invention of claim 21. The Examiner cites col. 8, lines 9-15 of Judge, which indicates that when it is determined that <u>data class objects</u> or instances of data objects are to be unloaded due to a low- or no-memory situation, the Data Manager removes references to the objects that are chosen to be unloaded. Col. 7, lines 33-50 indicate that a request may be received from an application, such as to unload a <u>data class</u>. However, Judge neither discloses nor suggests communicating <u>that the application has decided to</u>

terminate and has entered a destroyed state from a loaded state, a paused state, or an active state. Moreover, Judge fails to disclose or suggest communicating that the application has decided to pause its execution and has entered the paused state from the active state. In other words, the application of claim 21 may choose to terminate its execution of pause its execution. Accordingly, Applicant respectfully submits that Judge fails to anticipate the invention of claim 21.

Claim 22 recites: "The computer program product as recited in claim 21, further comprising:

instructions for communicating that the application wishes to resume execution and enter the active state from the paused state." Similarly, claim 26 recites, in relevant part, "instructions for communicating that the application wishes to resume execution and enter the active state from the paused state."

Col. 4, lines 11-19 of Judge merely indicate that executing applications interact with data objects. Judge fails to disclose or suggest communicating that the application wishes to resume execution and enter the active state from the paused state. In other words, the application of claims 22 and 26 may communicate that it wishes to resume its execution. Judge neither discloses nor suggests the invention of claim 22 or 26. Accordingly, Applicant respectfully submits that claims 22 and 26 are allowable over the cited art.

Claim 27 recites: "The computer program product as recited in claim 26, further comprising:

instructions for communicating that the application has decided to pause its execution and has entered the paused state from the active state." Col. 5, lines 40-45 indicate that data classes are unloaded by the data manager, not that the application is unloaded (or paused). Moreover, Judge fails to disclose or suggest that the application has elected to pause its own execution. Accordingly, Applicant respectfully submits that claim 27 is allowable over the cited art.

Claim 29 recites: "The computer program product as recited in claim 28, further comprising:

instructions for communicating that the application has decided to terminate and has entered the destroyed state." While col. 7, lines 27-40 indicate that a data manager may receive a request (e.g., load data class) from an application, Judge fails to disclose that the application has elected to terminate its own execution. Accordingly, Applicant respectfully submits that claim 29 is allowable over the cited art.

Applicant believes that the independent claims and dependent claims are allowable for the reasons previously set forth. The dependent claims depend from one of the independent claims and are therefore patentable over the cited art for at least the same reasons. However, the dependent claims recite additional limitations that further distinguish them from the cited references. Hence, it is submitted that the dependent claims are patentable over the cited art. The additional limitations recited in the independent claims or the dependent claims are not further discussed as the above discussed limitations are clearly sufficient to distinguish the claimed invention from the cited art. Thus, it is respectfully requested that the Examiner withdraw the rejection of the claims under 35 USC §102.

If there are any issues remaining which the Examiner believes could be resolved through either a Supplemental Response or an Examiner's Amendment, the Examiner is respectfully requested to contact the undersigned attorney at the telephone number listed below.

Applicants hereby petition for an extension of time which may be required to maintain the pendency of this case, and any required fee for such extension or any further fee required in connection with the filing of this Amendment is to be charged to Deposit Account No. 50-0388 (Order No. SUN1P507).

Respectfully submitted,
BEYER, WEAVER & THOMAS, LLP

[illegible]

Elise R. Heilbrunn
Reg. No. 42,649

BEYER, WEAVER & THOMAS, LLP
P.O. Box 778
Berkeley, California 94704-0778
Tel. (510) 843-6200